In The Claims:

1. (currently amended) A computer program product for digitally notarizing a collection comprising a plurality of data streams, the computer program product embodied on one or more computer-usable media and comprising:

computer-readable program code ~~means for computing~~ configured to compute a hash value over each of the plurality of data streams, wherein each data stream is created by a different application processing component;

computer-readable program code ~~means for combining~~ configured to combine each hash value for each of the plurality of data streams with a unique identifier of the application processing component which created the data stream for which the hash value was computed, thereby creating a combination data block;

computer-readable program code ~~means for hashing~~ configured to hash the combination data block;

computer-readable program code ~~means for digitally signing~~ configured to digitally sign the hashed combination data block with a private cryptographic key, wherein the private cryptographic key and a public cryptographic key which is cryptographically associated therewith represent a digital notary; and

computer-readable program code ~~means for providing~~ configured to provide the digitally signed hashed combination data block, along with the combination data block, as the digital notarization for the collection plurality of data streams, wherein the digital notarization cryptographically seals contents of the collection of data streams.

2. (currently amended) The computer program product according to claim 1, wherein:

the computer-readable program code ~~means for combining~~ configured to combine and the computer-readable program code ~~means for hashing~~ configured to hash operate on pairs of (hash values, identifiers), one pair for each of the plurality of data streams;

the computer-readable program code ~~means for digitally signing~~ configured to digitally sign digitally signs each of the hashed pairs; and

the computer-readable program code ~~means for providing~~ configured to provide provides the digitally signed hashed pairs, along with the hashed pairs, as the digital notarization.

3. (currently amended) The computer program product according to claim 1, wherein:

the computer-readable program code ~~means for computing~~ configured to compute a hash operates periodically, upon expiration of an elapsed time value, to compute a hash value over each of a plurality of segments of each data stream;

the computer-readable program code ~~means for combining~~ configured to combine, the computer-readable program code ~~means for hashing~~ configured to hash, and the computer-readable program code ~~means for digitally signing~~ configured to digitally sign all operate on the periodically-computed hash values for each data stream; and

the computer-readable program code ~~means for providing~~ configured to provide provides the digitally signed periodically-computed hash values, along with the periodically-computed hash values, as the digital notarization; and

further comprising computer-readable program code ~~means for inserting~~ configured to insert an identification of a time corresponding to each of the periodically-computed hash values at appropriate locations within each of the data streams.

4. (currently amended) The computer program product according to claim 3, wherein the computer-readable program code ~~means for inserting~~ configured to insert uses MPEG-4 synchronization timestamping.

5. (original) The computer program product according to claim 3, wherein authenticity and integrity of each of the segments is independently verifiable.

6. (currently amended) The computer program product according to claim 3, further comprising:

computer-readable program code ~~means for extracting~~ configured to extract selected ones of the segments of the data streams; and

computer-readable program code ~~means for verifying~~ configured to verify integrity of the extracted selected ones using the public cryptographic key of the digital notary.

7. (currently amended) The computer program product according to claim 3, further comprising:

computer-readable program code ~~means for authenticating~~ configured to authenticate, by the digital notary, each of the application processing components;

computer-readable program code ~~means for extracting~~ configured to extract selected ones of the segments of the data streams; and

computer-readable program code ~~means for verifying~~ configured to verify authenticity of the extracted selected ones using the public cryptographic key of the digital notary and the digital notarization.

8. (currently amended) The computer program product according to claim 1, further comprising:

computer-readable program code ~~means for adding~~ configured to add an additional data stream to the collection, wherein the additional data stream comprises the digital notarization.

9. (original) The computer program product according to claim 7, wherein the identifiers serve to identify data streams from each of the authenticated application processing components.

10. (currently amended) The computer program product according to claim 1, further comprising computer-readable program code ~~means for authenticating~~ configured to authenticate each of the application processing components using the unique identifier thereof, along with a digital signature of the unique identifier that is created using a private key of the application processing component.

11. (original) The computer program product according to claim 10, wherein inclusion of the unique identifiers within the combination data block allows concluding that each data stream in the collection was created by an authentic application processing component if operation of a verification process succeeds, wherein the verification process further comprises:

using the public cryptographic key of the digital notary to decrypt the digitally signed hashed combination data block, yielding a new version of the hashed combination data block and a new version of the combination data block;

computing a new hash over the new version of the combination data block; and

determining whether the new hash is identical to the new version of the hashed combination data block.

12. (original) The computer program product according to claim 11, wherein successful operation of the verification process also allows concluding that the data streams in the collection have not been altered.

13. (currently amended) A system for digitally notarizing a collection comprising a plurality of data streams, comprising:

means for computing a hash value over each of the plurality of data streams, wherein each data stream is created by a different application processing component;

means for combining each hash value for each of the plurality of data streams with a unique identifier of the application processing component which created the data stream for which the hash value was computed, thereby creating a combination data block;

means for hashing the combination data block;

means for digitally signing the hashed combination data block with a private cryptographic key, wherein the private cryptographic key and a public cryptographic key which is cryptographically associated therewith represent a digital notary; and

means for providing the digitally signed hashed combination data block, along with the combination data block, as the digital notarization for the collection of data streams, wherein the digital notarization cryptographically seals contents of the collection of data streams.

14. (original) The system according to claim 13, wherein:

the means for combining and the means for hashing operate on pairs of (hash values, identifiers), one pair for each of the plurality of data streams;

the means for digitally signing digitally signs each of the hashed pairs; and

the means for providing provides the digitally signed hashed pairs, along with the hashed pairs, as the digital notarization.

15. (original) The system according to claim 13, wherein:

the means for computing a hash operates periodically, upon expiration of an elapsed time value, to compute a hash value over each of a plurality of segments of each data stream;

the means for combining, the means for hashing, and the means for digitally signing all operate on the periodically-computed hash values for each data stream; and

the means for providing provides the digitally signed periodically-computed hash values, along with the periodically-computed hash values, as the digital notarization; and

further comprising means for inserting an identification of a time corresponding to each of the periodically-computed hash values at appropriate locations within each of the data streams.

16. (original) The system according to claim 15, wherein the means for inserting uses MPEG-4 synchronization timestamping.

17. (original) The system according to claim 15, wherein integrity of each of the segments is independently verifiable.

18. (original) The system according to claim 15, further comprising:

means for extracting selected ones of the segments of the data streams; and

means for verifying integrity of the extracted selected ones using the public cryptographic key of the digital notary.

19. (original) The system according to claim 15, further comprising:

means for authenticating, by the digital notary, each of the application processing components;

means for extracting selected ones of the segments of the data streams; and

means for verifying authenticity of the extracted selected ones using the public cryptographic key of the digital notary and the digital notarization.

20. (original) The system according to claim 13, further comprising means for adding an additional data stream to the collection, wherein the additional data stream comprises the digital notarization.

21. (original) The system according to claim 19, wherein the identifiers serve to identify data streams from each of the authenticated application processing components.

22. (original) The system according to claim 13, further comprising means for authenticating each of the application processing components using the unique identifier thereof, along with a digital signature of the unique identifier that is created using a private key of the application processing component.

23. (original) The system according to claim 22, wherein inclusion of the unique identifiers within the combination data block allows concluding that each data stream in the collection was created by an authentic application processing component if operation of a verification process succeeds, wherein the verification process further comprises:

using the public cryptographic key of the digital notary to decrypt the digitally signed hashed combination data block, yielding a new version of the hashed combination data block and a new version of the combination data block;

computing a new hash over the new version of the combination data block; and

determining whether the new hash is identical to the new version of the hashed combination data block.

24. (original) The system according to claim 23, wherein successful operation of the verification process also allows concluding that the data streams in the collection have not been altered.

25. (currently amended) A method of digitally notarizing a collection comprising a plurality of data streams, comprising: ~~comprising steps of:~~

computing a hash value over each of the plurality of data streams, wherein each data stream is created by a different application processing component;

combining each hash value for each of the plurality of data streams with a unique identifier of the application processing component which created the data stream for which the hash value was computed, thereby creating a combination data block;

hashing the combination data block;

digitally signing the hashed combination data block with a private cryptographic key, wherein the private cryptographic key and a public cryptographic key which is cryptographically associated therewith represent a digital notary; and

providing the digitally signed hashed combination data block, along with the combination data block, as the digital notarization for the collection of data streams, wherein the digital notarization cryptographically seals contents of the collection of data streams.

26. (currently amended) The method according to claim 25, wherein:

the combining ~~step~~ and the hashing ~~step~~ operate on pairs of (hash values, identifiers), one pair for each of the plurality of data streams;

the digitally signing ~~step~~ digitally signs each of the hashed pairs; and

the providing ~~step~~ provides the digitally signed hashed pairs, along with the hashed pairs, as the digital notarization.

27. (currently amended) The method according to claim 25, wherein:

~~the step of~~ computing a hash operates periodically, upon expiration of an elapsed time value, to compute a hash value over each of a plurality of segments of each data stream;

~~the~~ combining ~~step~~, ~~the~~ hashing ~~step~~, and ~~the~~ digitally signing ~~step~~ all operate on the periodically-computed hash values for each data stream; and

~~the~~ providing ~~step~~ provides the digitally signed periodically-computed hash values, along with the periodically-computed hash values, as the digital notarization; and

further comprising ~~the step of~~ inserting an identification of a time corresponding to each of the periodically-computed hash values at appropriate locations within each of the data streams.

28. (currently amended) The method according to claim 27, wherein ~~the~~ inserting ~~step~~ uses MPEG-4 synchronization timestamping.

29. (original) The method according to claim 27, wherein integrity of each of the segments is independently verifiable.

30. (currently amended) The method according to claim 27, further comprising: ~~comprising the steps of:~~

extracting selected ones of the segments of the data streams; and

verifying integrity of the extracted selected ones using the public cryptographic key of the digital notary.

31. (currently amended) The method according to claim 27, further comprising: ~~comprising the steps of:~~

authenticating, by the digital notary, each of the application processing components;

extracting selected ones of the segments of the data streams; and

verifying authenticity of the extracted selected ones using the public cryptographic key of the digital notary and the digital notarization.

32. (currently amended) The method according to claim 25, further comprising ~~the step of~~ adding an additional data stream to the collection, wherein the additional data stream comprises the digital notarization.

33. (original) The method according to claim 31, wherein the identifiers serve to identify data streams from each of the authenticated application processing components.

34. (currently amended) The method according to claim 25, further comprising ~~the step of~~ authenticating each of the application processing components using the unique identifier thereof, along with a digital signature of the unique identifier that is created using a private key of the application processing component.

35. (original) The method according to claim 34, wherein inclusion of the unique identifiers within the combination data block allows concluding that each data stream in the collection was created by an authentic application processing component if operation of a verification process succeeds, wherein the verification process further comprises:

using the public cryptographic key of the digital notary to decrypt the digitally signed hashed combination data block, yielding a new version of the hashed combination data block and a new version of the combination data block;

computing a new hash over the new version of the combination data block; and

determining whether the new hash is identical to the new version of the hashed combination data block.

36. (original) The method according to claim 35, wherein successful operation of the verification process also allows concluding that the data streams in the collection have not been altered.

37. (currently amended) A digitally notarized collection of data streams, comprising:

a plurality of data streams in the collection, wherein each data stream is created by a different application processing component; and

a digital notarization of the collection, created by ~~the steps of~~:

computing a hash value over each of ~~each of~~ the plurality of data streams;

combining each hash value <u>for each of the plurality of data streams</u> with a unique identifier of the application processing component which created the data stream for which the hash value was computed, thereby creating a combination data block;

hashing the combination data block;

digitally signing the hashed combination data block with a private cryptographic key, wherein the private cryptographic key and a public cryptographic key which is cryptographically associated therewith represent a digital notary; and

providing the digitally signed hashed combination data block, along with the combination data block, as the digital notarization for the collection of data streams, wherein the digital notarization cryptographically seals contents of the collection of data streams.

38. (currently amended) A method of doing business using digitally notarized data streams, comprising ~~steps of~~:

digitally notarizing a collection comprising a plurality of data streams, further comprising ~~steps of~~:

computing a hash value over each of the plurality of data streams, wherein each data stream is created by a different application processing component;

combining each hash value for each of the plurality of data streams with a unique identifier of the application processing component which created the data stream for which the hash value was computed, thereby creating a combination data block;

hashing the combination data block;

digitally signing the hashed combination data block with a private cryptographic key, wherein the private cryptographic key and a public cryptographic key which is cryptographically associated therewith represent a digital notary; and

providing the digitally signed hashed combination data block, along with the combination data block, as the digital notarization for the collection of data streams; and

verifying authenticity of the digitally notarized collection of data streams, by a receiver of the digital notarization, further comprising:

using the public cryptographic key of the digital notary to decrypt the digitally signed hashed combination data block, yielding a new version of the hashed combination data block and a new version of the combination data block;

computing a new hash over the new version of the combination data block; and

determining whether the new hash is identical to the new version of the hashed combination data block, and if so, concluding that the data streams in the collection have not been altered.